

Data security and encryption in low-power Wireless Sensor Networks

Ioan Deaconu, Musat Andrei
Automatic Control and Computers Faculty
University Politehnica of Bucharest,
{ioan.deaconu@cti.pub.ro, andrei.musat@cti.pub.ro}

I. INTRODUCTION

Wireless Sensors are low cost, low power devices optimized to perform custom tasks. They usually gather information from their surroundings which can be sent towards base station servers in order to be stored and processed. This communication is generally achieved with the use of gateways. The gateway is usually connected to a more powerful device that can analyze the received informations and take certain actions based on the results. Often, the information transmitted by wireless sensors represents sensitive data. It is for this reason that security protocols are implemented to prevent attacks that can intercept, replicate or alter the data.

Currently, security protocols in wireless sensor networks use mostly key based encryption algorithms. While this method can achieve great efficiency in terms of data security and protection, it also requires a certain level of computational power and is not always a task which is quickly executed. More so, in order to use such protocols, nodes must store all the necessary keys. Due to their design, wireless sensor often do not possess the necessary resources. They seldom have external memories attached to them and their processing power is limited to microprocessors which run at frequencies in the range of 1-100 MHz.

Another limitation of using this type of protocols to encrypt data is related to energy consumption. Usually, these sensors are powered by small batteries with a limited capacity. If the microprocessor has to perform intensive computations, these batteries shall be drained in short amounts of time. Even equipping sensors with energy harvesting peripherals does not ensure that the battery lifespan is greatly increased.

The approach presented in this paper attempts to implement more simple encryption algorithms which, combined with hardware encryption methods, can achieve an acceptable level of data security while ensuring that power consumption is kept to a minimum.

The proposed method relies on using available AES ECB encryption in internode communication. Since nodes can both encrypt and decrypt messages using the ECB protocol, it can safely be used for messages which do not contain critical data, but identify each node in the network. Then, the data itself should be encrypted using AES CBC protocol. However,

because the hardware does not support CBC decryption, it will have to be implemented at the software level.

This approach tries to find the balance point in the trade-off between security and energy consumption. While the data might not be protected as well as when key based algorithms are used, the energy consumption will be minimized thus increasing the life span of the sensor.

II. RELATED WORK

Security is of prime importance in Wireless Sensor Networks. Nodes transfer important data between them and the cost of checking each transfer if it is valid and has not been altered or intercepted is very high considering both battery life and processing time.

Possible attacks that might subdue a Wireless Sensor Network include:

- Wormhole attack: the attacker sends the received messages from one part of the network in a different part of the network. As a result, the nodes from both areas considers that the other nodes are neighbours and vice-versa.
- Blackhole/Sinkhole attack: the attacker makes it self more appealing from a routing point of view in order to receive all the messages from the network.
- Sybil attack: the attacker assumes the identity of one or more valid sensors[6].
- Selective forwarding attack: the attacker is able to intercept messages and drop certain packets or forward them [3].
- Hello Flood Attack: the attacker uses HELLO packets to flood the neighbours in order to force the nodes to trust him.

No current security framework offers complete protection against all types of attacks, but they offer protection against certain attacks. All of these implementations rely on software encryption methods.

- SPINS - 2002 - The communication parties create independent keys for encryption and decryption and MAC keys for communication. It provides security against Data and Information Spoofing and Message Replay Attacks[7].

- LEAP - 2003 - The protocol employs that the nodes exchange more than one type of message between them, so the framework uses 4 different keys. It provides security against HELLO flood attack, Sybil attack and minimizes the consequences of spoofing, altering, replay routing information and selective forwarding attacks [9].
- TinySec - 2004 - The key is pre-deployed on the node, but it does not provide any solution for changing the key. If a node is compromised, the entire network will be compromised. It provides security against Data and Information Spoofing and Message Replay Attacks [4].
- LEAP+ - 2006 - It uses the same idea as LEAP, but the overhead is reduced. It provides security against Confidentiality and authentication, HELLO flood attack, Sybil attack and minimizes the consequences of spoofing, altering, replay routing information and selective forwarding attacks [9].
- MiniSec - 2007 - Uses a counter IV. The counter is incremented locally and only the last bits of the counter are sent. It provides security against Authentication, Data Secrecy and Reply Attack [5].
- pDCS - 2009 - It uses 5 different keys to achieve It provides security against Location and Query privacy [8].
- TinyKey - 2011 - An improvement of TinySec, adds the key management system, in order to be able to change the key after the node is deployed. It provides security against Message authentication, confidentiality and integrity [1].
- ERP-DCS - 2013 - It proposes a different way of creating and storing keys when compared with pDCS. It provides security against Location and Query privacy [2].

III. ARCHITECTURE

A. Hardware

The processing power and wireless capability of the wireless sensor nodes are provided by an Atmel ZigBit 900MHZ RF module. It contains an ATmega 1281V 8-bit microcontroller connected to an AT86RF212 RF Transceiver via a SPI interface. The ATmega 1281V is a low power 8 bit microcontroller that is connected to the onboard sensors of the node. In order to be able to transmit or secure the data, the microcontroller will communicate with the RF Transceiver. The Transceiver controller is a very low power chip, capable of sending data up to 6 km. Also, the Transceiver contains a security module compatible with AES-128. It supports hardware encryption and decryption for AES 128 ECB, but but for the AES 128 CBC it is available only the hardware encryption.

B. Software

From the software perspective, the architecture is composed of two parts: gathering data from the sensors and encrypting it before transmitting it to another sensor of the central gateway. The encryption method for the data shall use both of the hardware supported methods: AES 128 ECB and AES 128

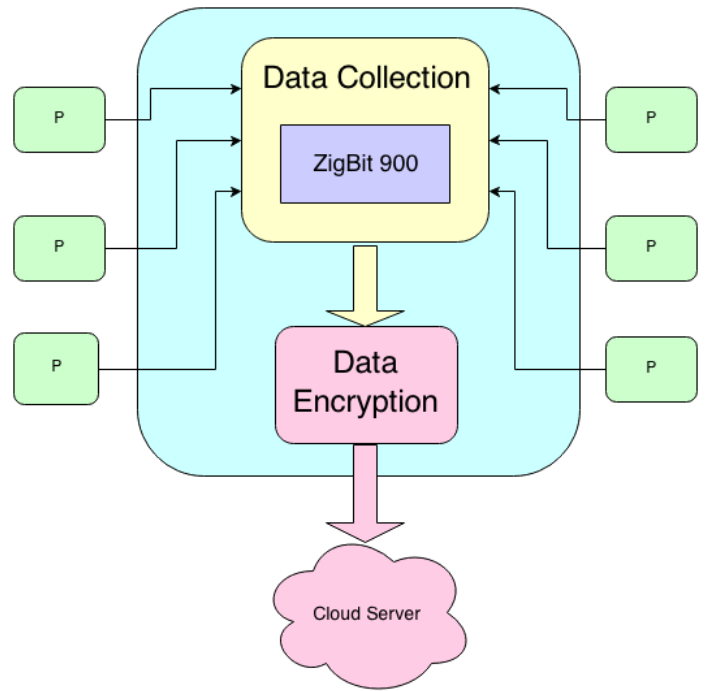


Figure 1. System Architecture

CBC. It is necessary to use them both because the nodes shall be transmitting two kinds of packets: a first type of packets which contains non-sensitive data and is used by the receiver to identify the sender and a second type of packets which contains the actual sensitive data. Since it supports both encryption and decryption, the first set of packets, those containing identification information, shall be encrypted using ECB. The data itself shall be encrypted using CBC. Since CBC decryption is not supported by default, a software CBC decryption implementation is necessary in order to allow the sensors to verify the data at regular intervals.

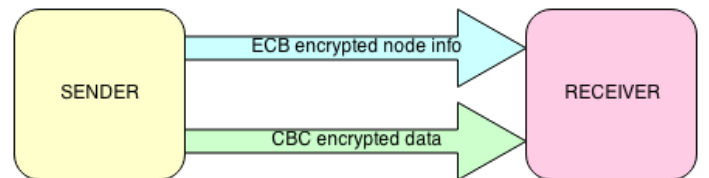


Figure 2. Packet Types

In addition to the CBC decryption, it is also required to have implementations of other software encryption algorithms in order to perform the performance analysis. The chosen algorithms which shall be used in benchmarking are Skipjack and RC5.

REFERENCES

- [1] R. Doriguzzi Corin, G. Russello, and E. Salvadori. Tinykey: A light-weight architecture for wireless sensor networks securing real-world applications. In *Wireless On-Demand Network Systems and Services*

- (WONS), *2011 Eighth International Conference on*, pages 68–75. IEEE, 2011.
- [2] J.-M. Huang, S.-B. Yang, and C.-L. Dai. An efficient key management scheme for data-centric storage wireless sensor networks. *IERI Procedia*, 4:25–31, 2013.
 - [3] S. Kaplantzis, A. Shilton, N. Mani, and Y. A. Sekercioglu. Detecting selective forwarding attacks in wireless sensor networks using support vector machines. In *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pages 335–340. IEEE, 2007.
 - [4] C. Karlof, N. Sastry, and D. Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 162–175. ACM, 2004.
 - [5] M. Luk, G. Mezzour, A. Perrig, and V. Gligor. Minisec: a secure sensor network communication architecture. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 479–488. ACM, 2007.
 - [6] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 259–268. ACM, 2004.
 - [7] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. E. Culler. Spins: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002.
 - [8] M. Shao, S. Zhu, W. Zhang, G. Cao, and Y. Yang. pdcs: Security and privacy support for data-centric sensor networks. *Mobile Computing, IEEE Transactions on*, 8(8):1023–1038, 2009.
 - [9] S. Zhu, S. Setia, and S. Jajodia. Leap+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(4):500–528, 2006.